



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Kerttu Myllyniemi

ARDUINO-PIANO

Tekniikka
2021

TIIVISTELMÄ

Tekijä	Kerttu Myllyniemi
Opinnäytetyön nimi	Arduino-Piano
Vuosi	2021
Kieli	suomi
Sivumäärä	25
Ohjaaja	Santiago Chavez

Opinnäytetyön tavoitteena on tehdä pianosta noin oktaavin kokoinen, eli kahdeksan koskettimen levyinen. Tällöin piano kulkeutuu mukana pienemmässäkin tilassa. Laite on suunniteltu siten, että se päästää ääntä ilman, että sitä itse soiteaan. Kytkenässä äänilähteenä käytetään summeria, ja äänen muodostuksessa taajuuksia.

Opinnäytetyö aloitettiin miettimällä mitä ja miten se tehdään. Työssä ensimmäisenä tehtiin kytkentä, jonka jälkeen lähdettiin suunnittelemaan ja kokoamaan koodia kasaan. Ensimmäisenä kirjoitettiin Arduinon järjestelmään ylös nuotit, koska melodia on helpompi hahmoittaa nuottien perusteella. Toiseksi annettiin nuoteille pituudet, jolloin melodia saatiin kuulostamaan järkevämmältä. Pituuksina käytettiin neljäsosanuotteja. Yksi neljäsosanuotti vastaa sekunneissa 0,67 s. Tätä testattiin toistamalla koodia Arduinolla toistuvasti.

Arduino Piano soitti kappaleita kuten pitikin. Projektiin olisi voinut lisätä kosketusantureita, jotta siitä olisi saanut täydemmän. Koodissa tärkeimpänä on melodia sekä nuottien kestot, sillä niiden piti täsmätä toisiinsa.

Avainsanat piano, Arduino, sähköpiano, kosketinsoittimet, soittimet

Piano

Author	Kerttu Myllyniemi
Title	Arduino Piano
Year	2021
Language	Finnish
Pages	25
Name of Supervisor	Santiago Chavez

The target of the thesis is to make the piano about the size of an octave, or eight keys. In this case, the piano travels in a smaller space. This device is also designed by the fact that encoding allows it to emit sound or a song by using different frequencies without being played by itself.

The project starts with planning what and how to do it. The first step was setting up the Arduino after which is planning and compiling the code. Notes were first written up in Arduino because melody is easier to perceive based on the notes. Second, the notes were given lengths to make the melody sound more sensible. Quarter notes were used as lengths. One quarter note corresponds to one bit which is about 0.67 seconds. The device was tested several times to get the duration of the notes match perfectly.

Arduino Piano worked as intended, it played songs as wanted. Capacitive measurement sensors could have been added to this project, to make it more complete.

Keywords piano, Arduino, electric piano, keyboard instruments,
instruments

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

SELITTEET

1	JOHDANTO.....	6
1.1	Piano	6
1.2	Tavoite	7
1.3	Työtavat	8
2	OPINNÄYTETYÖPROSESSI	9
2.1	Alku 9	
2.2	Vaatimuksia.....	9
2.3	Arduino Uno.....	9
2.3.1	Mikä on Arduino?.....	10
2.3.2	Miksi Arduino?	10
2.4	Kytkentä	12
3	KOODIT	14
3.1	Pitches.h tiedosto	14
3.2	Merkinnät.....	16
3.3	Hopeinen kuu.....	17
3.3.1	Hopeinen Kuu koodi.....	18
3.3.2	Banana Fish – Prayer X.....	20
4	TESTAUS JA TULOKSET.....	22
4.1	Testaus	22
4.2	Tulokset.....	22
4.3	Kehitysideoita	23
4.4	Oman osaamisen arviointi	Virhe. Kirjanmerkkiä ei ole määritetty.
	LÄHTEET	24

SELITTEET

Arduino = Mikro-ohjain, joka perustuu avoimeen laitteistoon

LED = hohtodiodi (Light-emitting diode)

Kytkenä = Usemman komponentin käyttö yhdessä

Koodi = Komentoja sisältävä käsky laitteelle tehdä haluttu asia

1 JOHDANTO

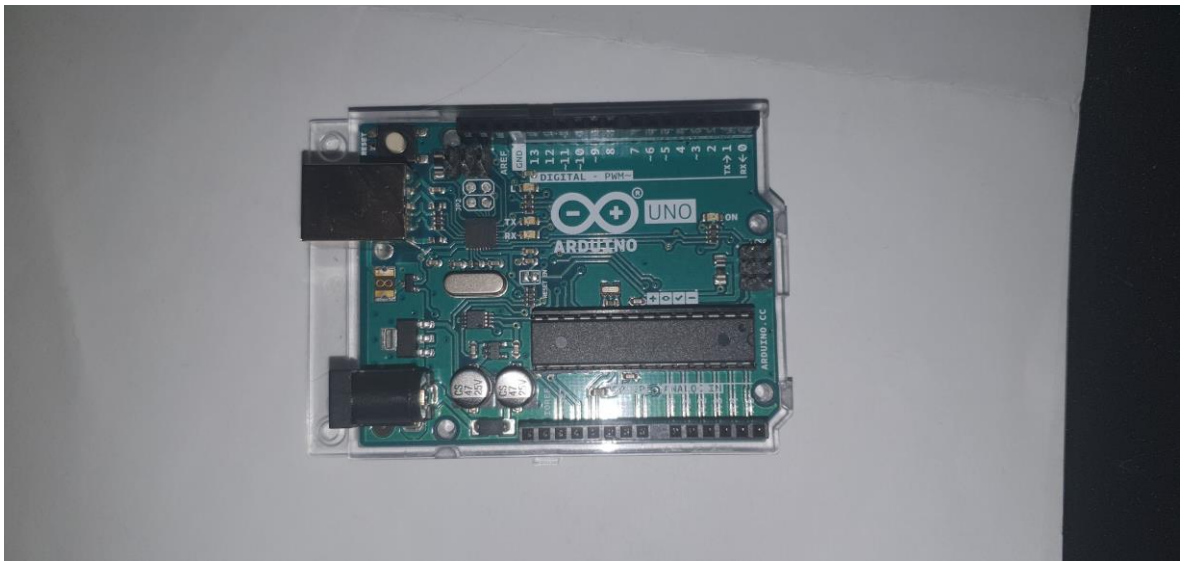
Opinnäytetyössä tehdään Arduino -pohjainen äänentoistolaite, joka muodostetaan kytkennästä sekä koodeista. Arduino -pianosta on tarkoitus tehdä pienempi kuin normaalista pianosta. Ydintavoitteena on saada se toistamaan ääntä, jotka sille on koodattu. Koodissa käytetään taajuustiedostoa pitches.h, jonka avulla itse koodista saadaan lyhyempi ja helpommin muokattava.

Työn alussa tärkeimmässä osassa oli kytkentä, sillä ilman sitä ei opinnäytetyöstä tulisi mitään. Vasta kytkennän jälkeen suoritetaan koodin kirjoitus ja muokaus.

1.1 Piano

Piano on instrumentti, tarkemmin sanottuna kosketinsoitin. Pianoja on kahta eri soitinta, sähköistä sekä akustista. Sitä voidaan soittaa yksin soolona tai bändissä muiden instrumenttien kanssa. Pianon koskettimien määrä on 88 ja ääniala on $7\frac{1}{4}$ oktaavia (Musiikki Wiki, 2009).

Normaali akustinen piano on leveydeltään 148 cm (F-Musiikki,2015) ja sen paino voi olla jopa 200 kiloa. Akustinen piano voi maksaa useita tuhansia euroja. Harvalla opiskelijalla on tähän varaa, jolloin Arduinomallinen piano voi tulla hyvinkin käytäntöön. Kuvassa 1 Arduino, jolla tämä opinnäytetyö on toteutettu.



Kuva 1. Arduino- alusta

1.2 Tavoite

Opinnäytetyön tavoitteenani oli onnistua käyttämään koulussa opittuja taitojani niin Arduinon, kuin koodauksenkin suhteen. Tarkoituksena ei ole oikea, realistinen sähköpiano, vaan saada siitä kopio Arduinolle. Pohjana työlleni käytin tietojani ja taitojani, joita on kertynyt vuosien aikana pianonsoittoharrastukseni ansiosta. Käytössäni oli sähköpiano (**Kuva 2.**) Pianon avulla sain taajuudet oikein, ja nuotit koodiin kohdalleen.

Toisena tavoitteenani oli saada opinnäytetyöstä yksinkertainen kytkennän sekä koodin osalta, jotta kuka tahansa sitä osaisi muokata tarvittaessa.



Kuva 2. Sähköpiano (kuva netistä lainattu)

1.3 Työtavat

Tietoja projektiin hain erilaisilta internet-foorumeilta. Koodiin sain apua netistä mm. pitches.h tiedoston suhteen. Itse .c tiedoston koodi on päällisin puolin itse kirjoitettu. Osien hankintaan käytin apunani sivustoa, jossa oli tehty vastaanvan tyylinen piano Arduinolla. Lähteenäni käytin dokumentin lopussa liitteenä olevan "Banana Pianon" osalistaa (Arduino Project Hub,2018).

Piano Arduinolla sujui suunnitellusti, pieniä ongelmia koodin sekä summerin toimivuuden kanssa tuli, mutta nämä ratkottiin sujuvasti. Tässä työssä tarvittiin hyviä ongelmanratkaisutaitoja sekä luovuutta.

2 OPINNÄYTETYÖPROSESSI

2.1 Alku

Opinnäytetyöprosessi aloitettiin tammikuussa opinnäytetyön suunnitelmalla, jossa pohjustettiin opinnäytetyön rakennetta. Suunnitelma sisältää opinnäytetyön tavoitellun aikataulutetun järjestyksen. Se alkaa opinnäytetyöpohjan tallennuksella. Tavoitteena oli kirjata ensin johdanto, tekniset tiedot sekä työn toteutus. Koodi sekä testaus tehtiin jälkepäin, kun kytkentä oli tehty ja valmis testattavaksi. Suunnitelmaan kuului myös kytkennälle tarvittavien osien hankinta.

Osat tilattiin opettajalta suunnitelman teon jälkeen. Osien saavuttua pystyi aloittamaan kytkennän sekä koodin kirjoituksen. Opinnäytetyötä kirjoitettiin aina, kun opinnäytetyöhön saatiin lisää materiaalia. Kuvia ja videomateriaalia kytkennän toimivuudesta tallennettiin aina, kun siihen oli mahdollisuus. Kun materiaalia löytyi lisää, pystyi opinnäytetyön kirjoittamaan loppuun.

2.2 Vaatimuksia

Vähimmäisvaatimuksena työlle oli saada koodattua pianolle kappaleita, joita se itse soittaa. Koodia muokkaamalla, tai eri koodin Arduinoon ajamalla saadaan piano soittamaan eri kappaleita. Kappaleet on myös mahdollista laittaa peräkkäin, jos koodiin ne näin kirjoitetaan.

2.3 Arduino Uno

Työssä käytetään Arduino Unoa. Aluksi suunnitelmana oli Arduino Mega, mutta päädyttiin siihen, että Unollakin pärjää. Arduino on laadukas avoin lähdekoodillinen mikrokontrolleri, ja sopii monipuoliseen sekä laadukkaaseen suunnitteluun. Sillä koodaaminen on yksinkertaista ja helppoa sekä se sopii erittäin hyvin aloittelijallekin. Apua koodaukseen löytyy niin kirjoista kuin internetistä.

2.3.1 Mikä on Arduino?

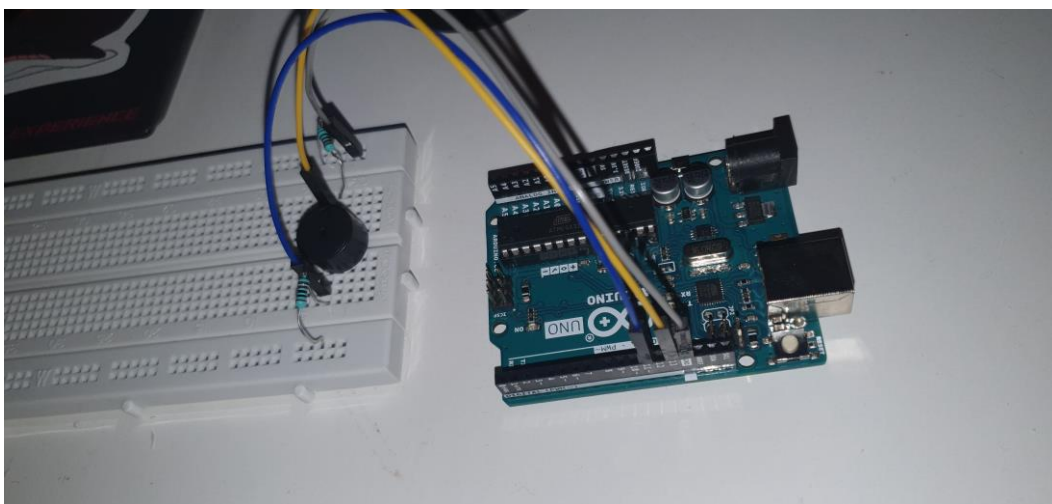
Arduino on avoimen lähdekoodin elektroniikka- alusta. Arduino syntyi Ivrea Interaction Design Institute- laitoksessa. Se perustettiin, jotta prototyyppien laatiminen olisi helppoa ja nopeaa. Sen käyttö perustuu yksinkertaisiin laitteisiin ja ohjelmiin. Arduino-levyt lukevat tulolähteet (Input) joko valoanturissa tai sormipainikkeessa ja muuttaa tämän lähdeksi (Output). Näin moottori aktivoituu ja sytyttää LED – valon. Arduinolle voi kertoa, mitä ja milloin se tekee lähettämällä tietoa ja ohjeita mikrokontrollerille. Tähän tarvitaan Arduinon ohjelma, jossa koodikieli on Arduinon oma. Tällä hetkellä ohjelmaversiona käytetään Arduino IDE 1.8.16. Ohjelman voi ladata Windowsille, Linuxille sekä Macia käyttäville laitteille (Arduino.cc, 2021 c)

2.3.2 Miksi Arduino?

Arduino on halpa, Arduino Starter Kit maksaa 79,90 euroa. Arduino Kittiin saa kuvassa 3 näkyvät osat. Osina saa mm. vastuksia, hyppylankoja, LED – näyttö jne. Arduinon ohjelmistoympäristö on helppo oppia ja ymmärtää. Arduinosta löytyy tietoa ympäri internetiä, lähes kaikkiin ongelmiin löytyy ratkaisu helposti. Se on erittäin hyvä aloittelijan käyttöön. Arduinoa voi käyttää usealla eri käyttöjärjestelmällä. (Arduino.cc, 2021 a)

2.4 Kytkenä

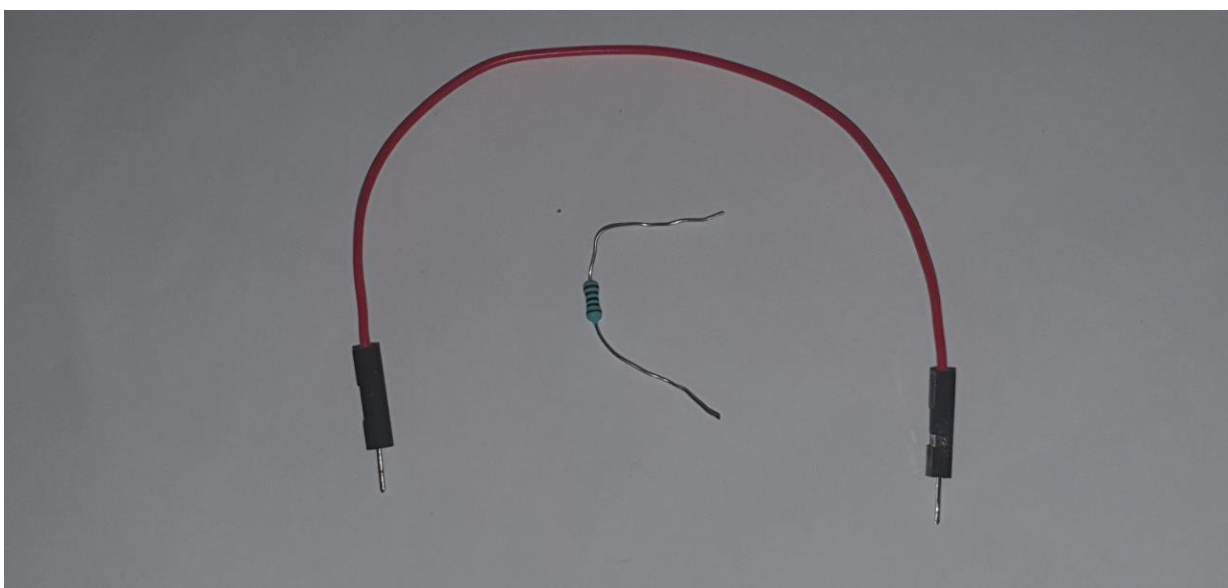
Laitteen kytkennästä halutaan yksinkertainen, tällöin sitä on helppo muokata ja muuttaa. Kuvassa 4 nähdään koko kytkentä. Jos kytkennästä olisi halunnut laajemman, ja siinä olisi käytetty esimerkiksi kosketusantureita, olisi tämä onnistunut usemman hyppylangan sekä vastuksen avulla. Kytkennässä käytetään taajuusbuzzeria BPT-14X (**Kuva 5.**), kahta 110 ohmin vastusta sekä hyppylankoja (**Kuva 6.**), joilla saadaan kytkentä tehtyä. Resistorien vastuksen suuruus vahvistettiin käyttämällä resistorin vahvuuslaskuria.



Kuva 4. Kytkenä



Kuva 5. Buzzer BPT-14X



Kuva 6. Vastus ja hyppylanka

3 KOODIT

Ensimmäisenä on pitches.h -tiedosto, jossa taajuudet määritellään. Jokainen taajuus on määritelty sen mukaan, miltä ne suurinpiirtein oikealla pianolla kuulostaa. Taajuudet on pyöristetty ylöspäin, sillä esimerkiksi NOTE_B0 taajuus olisi oikeasti 30,87 Hz, mutta 0,13 Hz erotus ei ole ihmiskorvalla kuultavaa. Ihmisen kuuloalue alkaa alhaisimmillaan 20 Hz, ja voi yltää jopa 20 000 Hz korkeuteen. Kuitenkin normaali kuuloalue on parhaimmillaan 2000 – 5000 Hz välillä. (NCBI 2001)

3.1 Pitches.h tiedosto

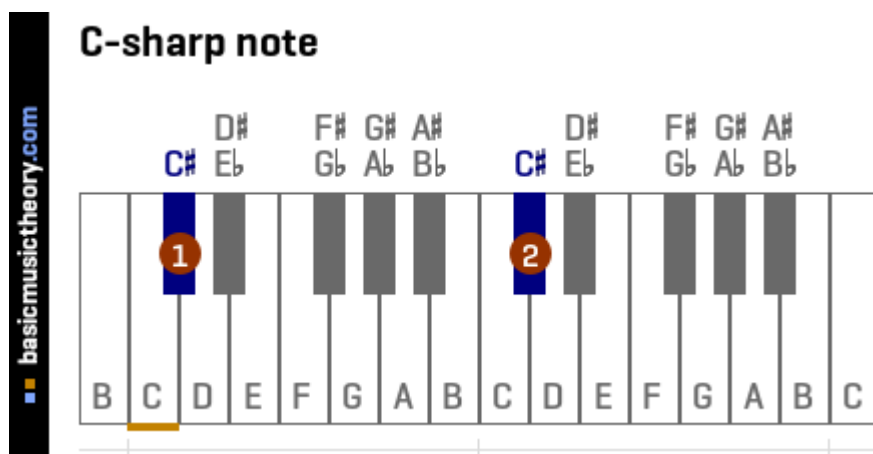
Koodissa #define tarkoittaa kuvausta, jolla annetaan nimi arvolle, tässä tapauksessa NOTE_X, ennen kuin koodi ajetaan. NOTE_X kertoo koodaajalle mistä nuotista on kyse. Numero jokaisen lauseen perässä vastaa taajuutta, jolla kyseinen nuotti soi.

```
#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 116
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165/*
```

```
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
```

3.2 Merkinnät

NOTE_C1 tarkoittaa näppäimistöä matalaa C – nuottia. NOTE_CS1 taas tarkoittaa koroitettua, eli nuoteissa kirjoitettu C#, kuten kuvasta 7 näkee.



Kuva 7. C-sharp nuotti

Tämä sama kaava toistuu nuoteille CDEFGAH. Kuvassa 7 esitetty nuotti B on ennen opetettu kirjaimena H, ja vain ylennetty H, eli H# olisi B, mutta ajat muuttuvat, ja tämänkin esitystapa on saattanut muuttua. Myös eri kielissä tämä muutos saattaa eritä.

3.3 Hopeinen kuu

Ohjeena ja mallina käytetään yksinkertaisia pianon nuotteja, kuten kuvassa 8 näkyy. Koodiin kirjoitettiin vain alku kappaleesta, sillä toimivuutta sekä esitystä varten, ei koko kappaletta olisi välttämättä järkeä koodata. Kappaleen valinnassa ja päätöksessä sain apua äidiltäni, sillä tämä kappale on lähellä hänen sydäntään.

Hopeinen kuu
Guarda che luna

Gualtiero Malgoni

Slowfox

Ho-pei-nen kuu luo me-rel-le sil-taa, ei tul-la
kos-kaan voi-si kai täl-lais-ta il-taa. O-do-tan
tur-haan si-mu-a, ar-main. Kuu kul-kee
pil-ven taak-se taas, on me-ri har-main. Kai-pa-us
vas-taa sy-dä-men ää-neen. On-ne-ni
tie-dän mä nyt taas niin yk-sin jää-neen. Toi-nen nyt
kuu-ta kans-sas kat-soo, toi-nen vie-nyt si-nut lie nyt. Ho-pei-nen
kuu luo me-rel-le sil-taa. Ho-pei-nen

Copyright © Southern Music Editions S.R.L.
Printed with permission of Edition Wilhelm Hansen AS, Copenhagen.

64

Kuva 8. Hopeinen Kuu nuotit

3.3.1 Hopeinen Kuu koodi

Koodissa nuotit ja nuottien kestot olen merkinnyt `/**/` väleillä, jotta muokkausvaiheessa on helpompi hahmottaa, missä kohti ollaan menossa, kun koodia testataan. Tämän takia koodissa ovat suuret välit ja koodin ulkoasu on muutoin ruma. Koodin ulkoasu ei juurikaan ole koodauksen kannalta hyväksyttävä, mutta pitkässä pötkössä siitä ei saanut muokkaus- sekä tarkistusvaiheissa mitään selvää.

```
#include "pitches.h" // lisätään .h tiedosto, missä taajuudet

#define ACTIVATED LOW // integroidun piirin IC nastan vaadittu
logiikka

const int PIEZO = 11;
const int LED = 13; //aktivoidaan ledi, kun tietoa menee perille

void setup()
{
  Serial.begin(9600); // 9600 bittiä sekunnissa
  digitalWrite(LED,LOW); //maa disabloitu
}

// melodian nuotit
int melody[] = {
NOTE_A1, NOTE_D2, NOTE_F2, /*D*/ NOTE_B2, NOTE_A2,/**/ NOTE_A1,
NOTE_D2, NOTE_F2, /*A*/ NOTE_B2, NOTE_A2, /**/ NOTE_A1, NOTE_CS2,
NOTE_E2,
/*D*/ NOTE_B2, /**/ NOTE_A2, NOTE_GS2, NOTE_A2,
/**/ NOTE_B2, /**/ NOTE_A2, NOTE_CS2, NOTE_B2, /*G*/ NOTE_A2,
NOTE_G2,/**/ NOTE_A1, NOTE_CS2, NOTE_E2,
/*A*/ NOTE_A2, NOTE_G2,/**/ NOTE_A1, NOTE_CS2, NOTE_E2,/**/
NOTE_G2, NOTE_F2,/**/ NOTE_D2, NOTE_E3,
NOTE_F2,/**/ NOTE_F2, NOTE_E2, NOTE_DS2, NOTE_E2,/**/ NOTE_F2,/**/
NOTE_E2, NOTE_G2, NOTE_F2,/**/ NOTE_F2, NOTE_E2,

};

// nuottien kesto: 4 = quarter note, 8 = eighth note, jne:
float noteDurations[] = {

2, 2, 2, /*D*/ 0.8, 0.8,/**/ 2, 2, 2, /*A*/ 0.8,
0.8, /**/ 2, 2, 2,
/*D*/ 0.8, /**/2, 2, 2,
/**/ 0.8,/**/ 2, 2, 2, /*G*/ 0.9, 0.8, /**/ 2,
2, 2,
/*A*/ 0.8, 0.8, /**/2, 2, 2, /*D*/ 0.8, 0.8, /**/ 2, 2, 2, /*E*/
0.8, 1.5, 2, 2, /**/ 0.8, /**/ 2, 2, 2, /*D*/ 1
,0.8,
```

```
};  
  
//looputon loop  
void loop(){  
  
    for (int thisNote=0; thisNote <46; thisNote++) {  
        int noteDuration = 600 / noteDurations[thisNote];  
        tone(11, melody[thisNote], noteDuration);  
        int pauseBetweenNotes = noteDuration * 1.5;  
        delay(pauseBetweenNotes);  
        noTone(11);  
  
    }  
  
    noTone(PIEZO);  
    digitalWrite(LED,LOW);  
    exit(0);  
}
```

3.3.2 Banana Fish – Prayer X

Toisena kappaleena koodasin Banana Fish -sarjassa kuuluvan loppusoiton Prayer

X.

```
#include "pitches.h"
#define ACTIVATED LOW

const int PIEZO = 11;
const int LED = 13;

void setup()
{
  Serial.begin(9600);
  digitalWrite(LED,LOW);
}

// notes in the melody:
int melody[] = {
  //eka säkeistö
  NOTE_B2, NOTE_DS3, NOTE_B3, NOTE_AS3, NOTE_B3, NOTE_FS3,
  NOTE_FS3, NOTE_E3, NOTE_DS3, NOTE_E3,
  NOTE_DS3,NOTE_CS3, NOTE_DS3, NOTE_E3, NOTE_FS3, NOTE_DS3,
  NOTE_GS3, NOTE_F3, NOTE_B3, NOTE_AS3,
  NOTE_GS3,NOTE_G3, NOTE_AS3, NOTE_DS3, NOTE_CS3, NOTE_B2, NOTE_DS3,
  NOTE_B3, NOTE_AS3, NOTE_B3,
  NOTE_B3, NOTE_FS3, NOTE_FS3,NOTE_E3, NOTE_DS3, NOTE_E3, NOTE_DS3,
  NOTE_CS3, NOTE_DS3, NOTE_E3,
  NOTE_FS3, NOTE_FS3, NOTE_DS3, NOTE_GS3,NOTE_F3, NOTE_B3, NOTE_AS3,
  NOTE_AS3, NOTE_AS3, NOTE_CS4,
  NOTE_AS3, NOTE_CS4, NOTE_AS3, NOTE_DS4, //0:31

  NOTE_DS3, NOTE_AS3, NOTE_GS3, NOTE_AS3, NOTE_FS3, NOTE_FS3,
  NOTE_GS3,
  NOTE_AS3, NOTE_GS3, NOTE_DS3, NOTE_FS3, NOTE_F3,
  NOTE_FS3, NOTE_GS3, NOTE_DS3, NOTE_AS3, NOTE_GS3, NOTE_AS3,
  NOTE_FS3,
  NOTE_FS3, NOTE_GS3, NOTE_AS3, NOTE_GS3, NOTE_DS3,

  NOTE_AS2, NOTE_GS2,0, NOTE_AS3, NOTE_AS3, NOTE_GS3, NOTE_AS3,
  NOTE_CS4,
  NOTE_AS3, NOTE_AS3, NOTE_DS3, NOTE_DS3, NOTE_F3, NOTE_FS3,
  NOTE_GS3, NOTE_AS3,
  NOTE_FS3, NOTE_FS3, NOTE_GS3, NOTE_AS3, NOTE_FS3, NOTE_F3,
  NOTE_F3,
  NOTE_FS3, NOTE_F3, NOTE_FS3, NOTE_GS3, NOTE_DS3,

  NOTE_F3, NOTE_FS3, NOTE_GS3, NOTE_FS3, NOTE_GS3, NOTE_FS3, NOTE_F3
  //0:53
```

```

};
// note durations: 4 = quarter note, 8 = eighth note, etc.:
float noteDurations[] = {

// eka säkeistö
2, 2, 2, 1, 1, 2, 2, 3, 2, 2, 1,
3, 3, 2, 3, 1, 2, 0.85, 2, 0.85, 2,
2, 2, 2, 2, 2, 2, 2, 1, 2, 2,
2, 2, 2, 2, 3, 1, 3, 3, 2, 3, 3,
1, 2, 0.85, 2, 0.85, 3, 2,
2, 3, 2, 3, 0.5,

2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1,
2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 3, 2, 2, 3, 2, 3, 2, 2, 1,
3, 2, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2,
2, 2, 2, 3, 2, 2, 2, 1.8

} ;

void loop(){

  for (int thisNote=0; thisNote <80; thisNote++) {
    int noteDuration = 600 / noteDurations[thisNote];
    tone(11, melody[thisNote], noteDuration);
    int pauseBetweenNotes = noteDuration * 1.5;
    delay(pauseBetweenNotes);
    noTone(11);

  }

  noTone(PIEZO);
  digitalWrite(LED,LOW);
  exit(0);

}

```

4 TESTAUS JA TULOKSET

4.1 Testaus

Koodia testattiin aluksi vain testaamalla ääntä, joko se tuli tai sitä ei tullut. Kun sitä tuli, piti kuunnella, että se on oikein. Jos ääni kuulosti huonolta tai väärältä, koodia muokattiin. Suurimpana ongelmana oli nuottien kestot. Kestojen selvittelyssä ja testauksessa menikin eniten aikaa koko työssä. Itse nuotit oli helppo kirjoittaa, sillä äänikorvaa löytyi, mutta nuottien kestot saattoivat olla esimerkiksi 0,25 neljäs-osanuottia väärässä eli noin 0,16 sekuntia. Tämä ei ole mikään kovin pitkä aika, mutta siihen kiinnittää kyllä huomiota kappaletta kuunnellessa.

Jossain vaiheessa summeri tuntui hajoavan, siitä ei kuulunut ääntä, vaikka LED-valo välkkyi. Tässä oli kyse siitä, ettei vastuksia ollut tarpeeksi. Kun aikansa niitä sääteli, niin huomasi, että ääni kuuluu vain, jos vastukseen koskee. Tässä taas oli kyse siitä, että + ja – johdot olivat väärinpäin. Loppujen lopuksi tässä oli vain koikeilu- ja virhemenetelmästä kyse, ennen kuin haluttuun lopputulokseen päästiin.

4.2 Tulokset

Ideani ja tulokseni täsmäsivät jotakuinkin toisiinsa. Halusin, että summeri pitää ääntä, kun sille annetaan koodi, jonka mukaan ääntä päästetään. Kun Arduinolle antoi käskyn ajaa koodi, lähetti se summerille käskyn soittaa koodatun kappaleen. Koodatut kappaleet soivat kuten haluttiin ja koodin muokkaus onnistuu tulevaisuudessakin, kun nuotit ja niiden kestot muutetaan halutuksi.

4.3 Kehitysideoita

Jos pianosta haluaisi lähes täysmittaisen, tarvittaisiin sitä varten Arduino Mega, jossa on peräti 54 input-output-pinniä. Summeri voisi olla parempilaatuinen, jolloin ääni olisi selkeämpi. Lisäksi näppäinten tilalle voisi hankkia kosketussensoreita, joita painamalla ääni aktivoituisi. Näin saataisiin pianollekin realistisempi ulkoasu ja toteutus. Myös taajuuksia voisi muokata tarkemmiksi, jolloin ääni olisi kirkkaampi ja tarkempi.

LÄHTEET

Arduino.cc, 2021 a. Arduino Starter Kit Multi-language. Viitattu 2.10.2021
<https://store.arduino.cc/products/arduino-starter-kit-multi-language>

Arduino.cc, 2021 b. digitalWrite(). Viitattu 7.10.2021 <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/>

Arduino.cc, 2021 c. What is Arduino? Viitattu 25.10.2021 <https://www.arduino.cc/en/guide/introduction>

Arduino Project Hub, 2018. Banana Piano. Viitattu 5.10.2021 <https://create.arduino.cc/projecthub/Baloo/banana-piano-5bb8b5>

Digi-Key. 4 Band Resistor Color Code Calculator. Viitattu 1.10.2021
<https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code>

F-Musiikki, 2015. B1PE Akustinen piano, musta kiiltävä. Viitattu 4.11.2021
<https://www.f-musiikki.fi/pianot/pystypianot/yamaha-pianob1pe-musta-kiiltava-piano-b1pe>

Learn.parallax.com. 2021. How the Hello Sketch Code Works. Viitattu 29.9.2021
<https://learn.parallax.com/tutorials/robot/shield-bot/robotics-board-education-shield-arduino/chapter-1-your-shield-bots-17>

Musiikki Wiki, 2009. Piano. 3.10.2021. <https://musiikki.fandom.com/wiki/Piano>

NCBI, 2001. The Audible Spectrum. Viitattu 6.11.2021
www.ncbi.nlm.nih.gov/books/NBK10924/

Vis, Peter. What Does Active LOW Mean? Viitattu 13.10.2021 <https://www.peter-vis.com/dictionary-of-digital-terms/what-does-active-low-mean/what-does-active-low-mean.html>

Youtube, 2020. Banana Fish Prayer X. Viitattu 31.10.2021 <https://www.youtube.com/watch?v=7BW9mKye6uE>

Youtube, 2010. Olavi Virta- Hopeinen Kuu. Viitattu 31.10.2021 <https://www.youtube.com/watch?v=WE5SrKzxYdY>